Universal Rate-Compatible LDPC Code Families for Any Increment Ordering

Sudarsan V. S. Ranganathan, Kasra Vakilinia, Dariush Divsalar*, and Richard D. Wesel
Department of Electrical Engineering, University of California, Los Angeles, Los Angeles, California 90095
*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109
Email: {sudarsanvsr, vakiliniak}@ucla.edu, Dariush.Divsalar@jpl.nasa.gov, wesel@g.ucla.edu

***Abstract*—Rate-compatible (RC) codes are at the core of systems with incremental redundancy. Usually, an RC code family supports successively lower code rates by sending specific increments of additional redundancy at each rate. That is, the order of the increments is fixed. However, in some multi-hop communication systems and also in recently proposed inter-frame coding, the order in which the decoder of the RC code receives the increments is not predetermined. A different ordering of the increments at the decoder may change the codes of various rates.**

This paper seeks RC codes that are universally good over all increment orderings. We call RC codes satisfying this requirement *universal for any increment ordering* (UIO) codes. We design protograph-based Raptor-like (PBRL) low-density parity-check (LDPC) code ensembles for UIO codes using protograph thresholds as components of two design metrics. One metric seeks codes that, at each code rate, have exactly the same frame error rate for all increment orderings. The other metric sacrifices strictly identical performance for every ordering to seek codes that achieve the best possible throughput in a variable-length setting with random increment ordering, as would occur with inter-frame coding. Simulation results of UIO-PBRL codes from the new ensembles show that our designs satisfy the two metrics.

I. Introduction

Low-density parity-check (LDPC) codes approach capacity with iterative decoding at sufficiently long block-lengths. First described as block codes by Gallager [1], they have since been designed for requirements such as rate-compatibility, where a single parity-check matrix is used at a set of predefined rates.

A recent work on rate-compatible LDPC (RC-LDPC) codes is that of Chen et al. [2]. Here, the authors introduce a class of codes called protograph-based Raptor-like (PBRL) LDPC codes, or simply PBRL codes. These RC codes are easily encodable and have excellent ensemble decoding thresholds and frame error rate (FER) performance. We refer the reader to [2] for a summary on the state-of-the-art in designing RC-LDPC codes, of which recent papers include [3], [4], [5].

Our work on RC-LDPC codes in this paper is motivated by Zeineddine and Mansour's recently proposed *inter-frame coding* [6], which works as follows: A certain number of message packets is to be transmitted. They are each coded separately using an RC code and the highest-rate parts are sent out. Along with this, linear combinations of increments are also transmitted, where each of the increments for a combination comes from the RC code for a different packet. At the receiver, whenever a channel packet is decoded successfully, all increments of its RC code become known and are used to reveal (from the linear combinations) new increments for the remaining packets. These new increments lower the rate of their corresponding RC-coded packets, hopefully allowing some of them to be decoded so that the process iterates and recovers all the message packets.

The decoder of the RC codes in an inter-frame code [6] is not guaranteed to see a specific ordering of its increments at each code rate. Therefore, in order to support a practical implementation of the scheme, which was not considered in [6], we need an RC code that performs well over all increment orderings. We call RC codes satisfying this requirement *universal for any increment ordering* (UIO) codes. It was noted in [6] that such codes had yet to be investigated in literature.

This paper designs UIO codes with PBRL LDPC structure via two design metrics. The first metric seeks codes that have, at each rate, the same FER performance for every increment ordering. This metric reflects the key assumption of the throughput analysis of inter-frame coding in [6] that at each rate the RC codes in an inter-frame code will perform exactly the same for any increment ordering. That is, their analysis considered only the number of increments received. Our design for this metric results in a threshold penalty of at most 0.4 dB compared to the original PBRL ensemble of Chen et al. [2] for rate-compatibility in the usual sense, which was designed to have the best threshold at each rate.

The second metric sacrifices identical performance for every increment ordering to obtain RC codes with the best possible throughput. Here the focus is on enabling early successful decoding by minimizing the average protograph threshold over all orderings at each rate. This led to the best throughput in simulations where increments are provided to the decoder in a random order, as would occur with inter-frame coding.

The paper is organized as follows: Section II reviews protographs and PBRL codes. Section III presents ensembles for UIO-PBRL codes designed for the two metrics and shows simulation results of the new UIO-PBRL codes satisfying the requirements of the metrics. Section IV concludes the paper.

Research is supported in part by NSF grants CCF-1162501 and CCF-1161822 (JPL Task Plan 82-17473). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. Research was carried out in part at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with NASA. This work used computational and storage services associated with the Hoffman2 Shared Cluster provided by UCLA Institute for Digital Research and Education's Research Technology Group.

II. Protographs and PBRL Codes

This section reviews protographs and PBRL codes. To begin with, a Tanner graph is a bipartite graph that corresponds to the parity-check matrix of an LDPC code. If the parity-check matrix is of size $m \times n$, then the Tanner graph has m check nodes and n variable nodes constituting the two parts.

Definition 1 (Protograph and protomatrix): A *protograph*, from Thorpe [7], is a small Tanner graph with n_c check nodes and n_v variable nodes. It may have multiple edges connecting a variable-check node pair. The biadjacency matrix of a protograph is usually called a *protomatrix*. We use the terms protograph and protomatrix interchangeably.

Definition 2 (Lifting): Lifting, by a *lifting factor* M, is a process that is applied to a protograph to yield a derived graph representing a large LDPC code. The protograph is first replicated M times, yielding M disconnected copies. Then, the connections to the check nodes of the set of M edges (across the M replicas) obtained from an edge of a variable-check node pair in the original protograph are permuted among the M copies of the corresponding variable-check node pair. The same is performed independently for every set of M edges.

Lifting is often performed in two steps if a protograph has multiple edges between some/all variable-check node pairs. The lifting factor chosen for the first step is small and is greater than or equal to the largest non-zero entry in the protomatrix. This is done to obtain an intermediate protomatrix with no multiple edges. The second lifting step, with a much bigger lifting factor, leads to the large derived graph.

If a protograph with n_v variable nodes has n_t that are "transmitted" and $n_v - n_t$ that are "punctured", we mean that all the M copies of a variable node in the resulting code are either transmitted or punctured according to their type in the protograph. The design rate of a protograph is $R \triangleq (n_v - n_c)/n_t$. Upon lifting, an LDPC code has Mn_c check nodes, Mn_v variable nodes, and a rate $r \geq R$. It also has the same degree distribution (see [8]) as the protograph.

Definition 3 (Ensemble): The set of all codes obtainable by lifting a protograph is called the *ensemble* of the protograph.

A. Protograph-based Raptor-like (PBRL) LDPC codes

A PBRL code ensemble [2] is defined by a protograph whose protomatrix is of the following structure:

$$\begin{bmatrix} H_{\text{HRC}} & 0 \\ H_{\text{IRC}} & I \end{bmatrix} \tag{1}$$

Here 0 and I represent all-zeros and identity matrices of appropriate dimensions respectively. The highest-rate code (HRC) part of the protograph, upon lifting, is structurally identical to the precode part of a Raptor code. Similarly, the degree-1 nodes connected to the check nodes in the incremental redundancy code (IRC) part are efficiently encoded as modulo-2 sums of the precode symbols in a manner similar to the Luby transform (LT) code in a Raptor code.

Fig. 1 shows an example PBRL protograph. Rate-compatibility starts with the highest-rate code. H_{IRC} lowers the rate as its degree-1 variable nodes are included one at a time. We refer the reader to Section II of Chen et al. [2] for a detailed introduction to PBRL codes.

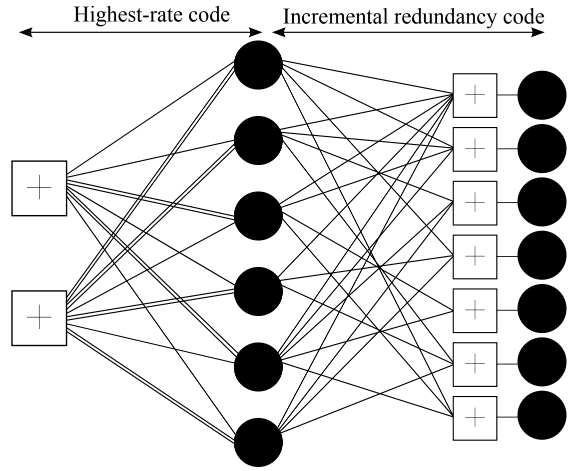

Fig. 1. A PBRL protograph (no punctured nodes) with a highest-rate code (HRC) of rate 2/3 and its incremental redundancy code (IRC). The IRC lowers the rate as its degree-1 variable nodes are included one at a time.

B. Design choices and remarks

We focus on design rates $8/i, i \in [10, 16]$. We design binary codes for binary-input additive white Gaussian noise channel (BI-AWGNC), with protograph iterative decoding thresholds computed using *reciprocal channel approximation (RCA)*. For a review of RCA as a one-dimensional approximation to density evolution of Richardson et al. [8], see Chen et al. [2] and Divsalar et al. [9]. We focus on long block-lengths (k = 16384 information bits) to obtain codes that operate fairly close to their ensemble thresholds. We assume that the incremental bits are delivered in chunks corresponding to each protograph variable node.

Remark 1: A PBRL ensemble is completely specified by H_{HRC}, and H_{IRC} or a combination of its rows. It is implicitly assumed that the overall protograph is formed by appropriately including the necessary 0 and I matrices. Also, the degree-1 variable nodes in H_{IRC} do not depend upon each other. This facilitates the design of UIO-PBRL codes for the first metric.

Remark 2: The threshold values in this work are the result of at least 1000 iterations of the RCA algorithm. Codes simulated are quasi-cyclic and have a first-step lifting factor of 4 and a second-step lifting factor of 512. First-step lifting used Hu et al.'s progressive edge-growth (PEG) algorithm [10], and second-step lifting used the circulant-PEG (C-PEG) algorithm. The ACE algorithm of Tian et al. [11] was also used in both steps. Simulation results shown were obtained using a maximum of 200 iterations of full-precision, flooding, LLR-domain belief propagation. At least 100 errors were collected for each FER point in any simulated E_b/N_0 vs. FER graph.

III. PBRL Ensembles for UIO-RC Codes

A. Long block-length PBRL ensemble of Chen et al. in [2]

A PBRL ensemble for rate-compatibility in the usual sense is designed as follows: First, we select an H_{HRC} with a degree

TABLE I
STATISTICS OF RCA THRESHOLDS (E_b/N_0) OVER BI-AWGNC CONSIDERING H_{HRC} AND VARIOUS COMBINATIONS OF ROWS OF H_{IRC} IN (2) FOR RATES $8/i, 10 \le i \le 16$. "SH" IS THE SHANNON LIMIT.

Rate	Sh (dB)	Min.	Max.	Ave.	Std. Dev.	Max-Min
8/10	2.04	2.179	2.179	2.179	0	0
8/11	1.459	1.579	2.044	1.809	0.16	0.465
8/12	1.059	1.199	1.897	1.49	0.195	0.698
8/13	0.762	0.897	1.528	1.172	0.174	0.631
8/14	0.53	0.662	1.153	0.86	0.135	0.491
8/15	0.342	0.462	0.668	0.568	0.083	0.206
8/16	0.187	0.308	0.308	0.308	0	0

distribution that has a good threshold and permits a low error floor. Then, each row of H_{IRC} is chosen by keeping all previous rows fixed and selecting edges for that row to obtain the best threshold possible while meeting constraints designed to preserve good error floor performance. The specific constraints and the complexity of the RCA algorithm dictate the overall complexity of this search. Note that, although this is a greedy search, one can obtain excellent thresholds and FER performance at all rates as demonstrated by Chen et al. in [2].

Let us first see how an original PBRL code from [2] behaves under different orderings of its increments. Consider the PBRL ensemble of [2] that was designed for long block-lengths with the best threshold at each rate. H_{HRC} and H_{IRC} ((13) and (14) in [2]), up to rate 8/16, are

$$H_{\text{HRC}} = \begin{bmatrix} 3&2&1&1&1&1&1&1&0&0&1 \\ 1&1&2&2&2&2&2&2&2&1&1 \\ 2&0&0&0&0&0&0&0&1&2&0 \end{bmatrix},$$
$$H_{\text{IRC}} = \begin{bmatrix} 2&1&2&1&0&0&0&0&1&0&0 \\ 1&2&1&0&1&0&0&0&0&1&0 \\ 2&0&0&2&0&1&0&0&0&0&0 \\ 1&1&0&0&0&0&1&0&0&2&0 \\ 2&2&0&0&0&0&0&1&0&0&0 \\ 1&0&0&0&0&2&0&0&0&2&0 \end{bmatrix}. \tag{2}$$

The design heuristics to obtain good thresholds for all code rates through the search in [2] led to the puncturing of the first protograph variable node in (2). We adopt the same principle in this work; our ensembles also have the first variable node punctured.

For the PBRL ensemble of (2), Table I shows the statistics of thresholds at each rate obtained by including various combinations of rows (corresponding to different orderings of the increments) of H_{IRC} with H_{HRC}. The number of ensembles at rate $8/(10+i), 0 \le i \le 6$ is $\binom{6}{i}$. From the table, we see a range of thresholds for all but the highest and lowest rates, with the maximum gap being 0.698 dB. The ensemble in (2) shows considerable variation in threshold for different orderings of the increments. Simulation results of the code obtained from this ensemble in [2] at rate 8/11, shown in Fig. 2, illustrate how the threshold variation manifests itself in FER performance.

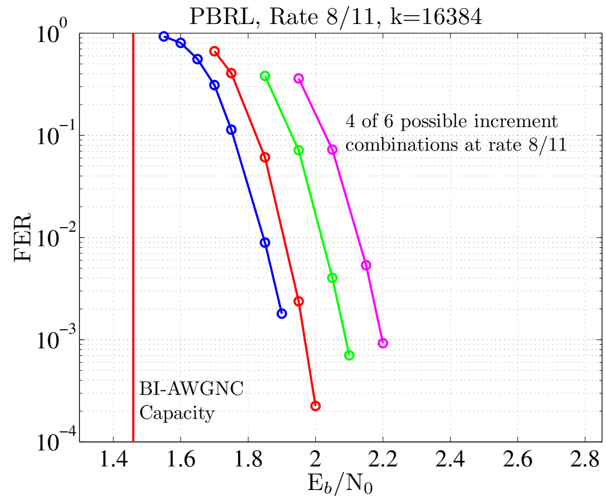

Fig. 2. Dependence of a PBRL code on the ordering of its increments

B. Design metric 1: UIO-PBRL codes that have, at each rate, identical FER performance for every ordering

This subsection seeks codes that have, at each rate, identical FER performance for every increment ordering. This metric requires that the thresholds of various ensembles at each rate formed using the different possible sets of rows of H_{IRC} be as close as possible to each other. This means that the rows of H_{IRC} need to be chosen such that the threshold gap at each rate (as in Table I) is as small as possible. Also, to be considered a good ensemble, the minimum threshold at each rate should be as small as possible. A small threshold gap at each rate mandates some symmetry in the ensemble, and the minimum threshold at each rate depends upon the degree distribution.

We restrict the maximum non-zero value in H_{IRC} to be equal to 2 in order to reduce the search complexity. Given an H_{HRC}, initial attempts to produce an H_{IRC} included making sure that each row of H_{IRC} has the same weight and that all rows have the same number of different types of non-zero values. We restricted the first column of H_{IRC} to either have all ones or all twos. The all-twos designs led to codes with poor FER performance. Our initial exhaustive search led to the following:

$$H_{\text{HRC}} = \begin{bmatrix} 3&1&1&1&1&1&1&2&2&2&0 \\ 1&1&2&2&2&2&2&1&1&1&0 \\ 2&0&0&0&0&0&0&0&0&0&1 \end{bmatrix},$$
$$H_{\text{IRC}} = \begin{bmatrix} 1&0&0&0&0&1&1&1&1&1&0 \\ 1&0&0&0&1&0&1&1&1&1&0 \\ 1&0&0&0&1&1&0&1&1&1&0 \\ 1&0&0&0&1&1&1&0&1&1&0 \\ 1&0&0&0&1&1&1&1&0&1&0 \\ 1&0&0&0&1&1&1&1&1&0&0 \end{bmatrix}. \tag{3}$$

Notice that every row in H_{IRC} is "equivalent" in that within H_{IRC} two rows can be exchanged and the matrix is still the same with a simple column exchange thereafter. Table II shows the statistics of the ensemble in (3). The maximum of all threshold gaps is 0.01 dB compared with 0.698 dB in Table I.

TABLE II
STATISTICS OF RCA THRESHOLDS (E_b/N_0) FOR ENSEMBLE IN (3).

Rate	Sh (dB)	Min.	Max.	Ave.	Std. Dev.	Max-Min
8/10	2.04	2.393	2.393	2.393	0	0
8/11	1.459	1.892	1.896	1.894	0.0022	0.004
8/12	1.059	1.494	1.502	1.498	0.0026	0.008
8/13	0.762	1.148	1.158	1.1539	0.0026	0.01
8/14	0.53	0.856	0.862	0.8596	0.002	0.006
8/15	0.342	0.640	0.642	0.641	0.0011	0.002
8/16	0.187	0.506	0.506	0.506	0	0

TABLE III
RCA THRESHOLDS (E_b/N_0, DECIBEL) FOR ENSEMBLE IN (4).

Rate	8/10	8/11	8/12	8/13	8/14	8/15	8/16
Sh	2.04	1.459	1.059	0.762	0.53	0.342	0.187
Thr.	2.462	1.934	1.518	1.156	0.842	0.606	0.474

Now, keeping H_{IRC} the same as in (3), we present an ensemble with zero gap between the thresholds at each rate:

$$H_{\text{HRC}} = \begin{bmatrix} 3&1&1&1&1&1&1&1&1&1&0 \\ 1&1&2&2&2&2&2&2&2&2&0 \\ 2&0&0&0&0&0&0&0&0&0&1 \end{bmatrix},$$

$$H_{\text{IRC}} = \begin{bmatrix} 1&0&0&0&0&1&1&1&1&1&0 \\ 1&0&0&0&1&0&1&1&1&1&0 \\ 1&0&0&0&1&1&0&1&1&1&0 \\ 1&0&0&0&1&1&1&0&1&1&0 \\ 1&0&0&0&1&1&1&1&0&1&0 \\ 1&0&0&0&1&1&1&1&1&0&0 \end{bmatrix}. \quad (4)$$

By the virtue of apparent symmetry in the ensemble in (4) (taking into consideration H_{HRC} and H_{IRC} together), it is clear that the threshold gap, Max.−Min., is zero at each rate. Table III presents the computed threshold at each rate.

Fig. 3 plots the gap to capacity of the original PBRL ensemble for long block-lengths in (2) and the ensemble in (4). Also plotted are the gaps to capacity of the worst-case threshold (Max.) for the PBRL ensembles in (2) and (3).

We simulated a code from the ensemble in (4), and the results are shown in Figs. 4 and 5. A zero threshold gap at each rate certainly seems to be a predictor of *exactly* the same performance for every increment ordering. We observed the same at every rate, and the results at other rates are not included for brevity. Note that the length of the code drawn from the ensemble may play a role in the actual performance due to local graph effects. We intend to investigate if similar results hold for much shorter block-lengths.

C. Design metric 2: UIO-PBRL codes with the best throughput over all increment orderings for inter-frame coding

The metric of minimizing threshold gap does not necessarily maximize the throughput in inter-frame coding. Our second design metric seeks codes that attempt to maximize throughput in inter-frame coding by decoding as early as possible. The

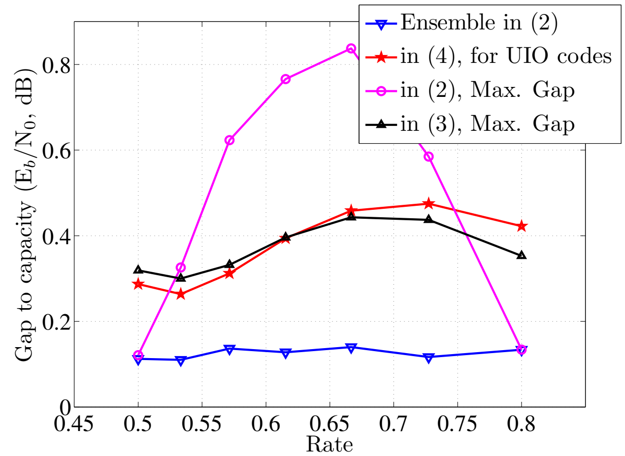

Fig. 3. Gap to BI-AWGNC capacity - Ensembles in (2), (3), (4)

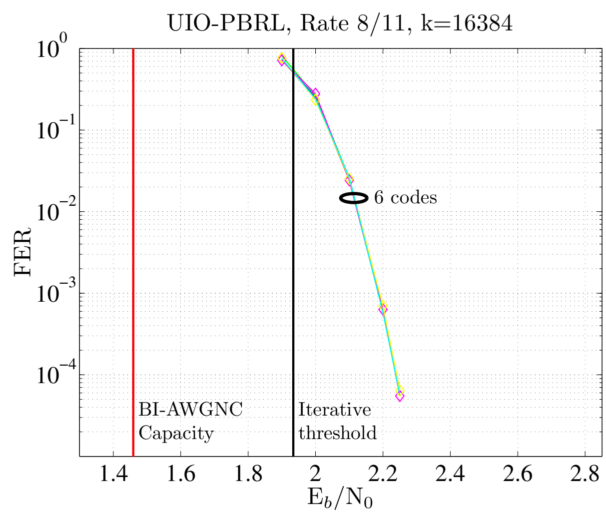

Fig. 4. Simulations of a UIO-PBRL code that show that the code has exactly the same performance irrespective of the ordering of its increments

simulations in this subsection are carried out according to how an RC code in an inter-frame code operates. The decoder starts decoding at the highest rate. If it is unsuccessful, a randomly chosen increment becomes available to the decoder. The process is repeated until the decoder decodes successfully or fails at the lowest rate. The throughput, in simulations, is the ratio of total number of information bits delivered successfully to the total number of codeword bits sent over the channel.

First, we compare the throughput of the original PBRL code for 16384 information bits (ensemble in (2)) and a UIO-PBRL code designed according to the first metric with the same H_{HRC} as in (2). The results, in Fig. 6, show that the original PBRL code has a higher throughput than the UIO-PBRL code at most channel SNRs. We do not provide the ensemble for this UIO-PBRL code for brevity. It has a maximum threshold gap of 0.02 dB and a gap of 0 dB at rates 8/10, 8/11, 8/15, and 8/16. As each increment ordering is equiprobable, one metric that affects throughput is the average threshold at each rate. We

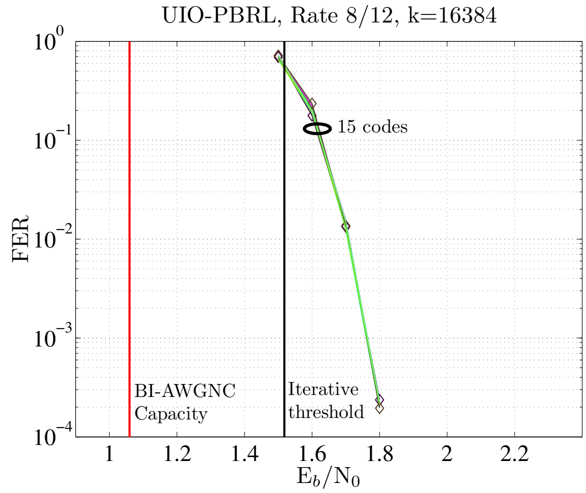

Fig. 5. Strictly identical performance at rate 8/12 for all 15 orderings

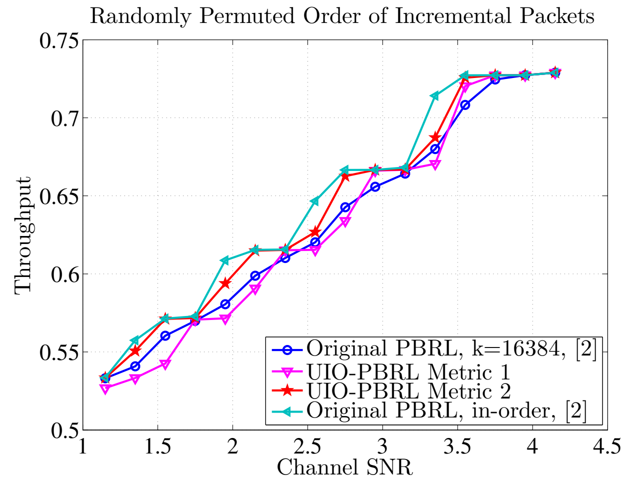

Fig. 6. Throughput comparison

found that the average threshold at each rate for the ensemble of the UIO-PBRL code is greater than or equal to the average threshold of the ensemble of the original PBRL code (Table I).

Based on this, we modified the PBRL design process to obtain an ensemble with low average thresholds. For this design, the H_{HRC} is the same as in (2). The original PBRL search designs H_{IRC} one row at a time to avoid exponentially high complexity. We retained this general procedure but widened our search space as follows: As each row is added, we considered the 3 best ensembles in terms of the threshold at that rate. That is, starting with the H_{HRC} at rate 8/10, we obtained 3 ensembles at rate 8/11, 9 at rate 8/12, and so on. We discarded isomorphic ensembles in obtaining the 3 best ensembles at each rate for a given matrix from the previous rate. From the resulting 729 ensembles, we obtained the ensemble in (5) with the best average threshold for all rates except 8/16 (Table IV).

$$H_{\text{IRC}} = \begin{bmatrix} 2&0&2&2&1&0&0&0&0&0&0 \\ 1&0&2&2&0&1&0&0&0&0&0 \\ 2&0&2&0&0&1&1&0&0&0&0 \\ 1&0&1&2&0&0&0&1&0&0&0 \\ 2&0&1&0&1&0&1&0&0&0&0 \\ 1&0&2&0&0&0&1&1&0&0&0 \end{bmatrix}. \quad (5)$$

Fig. 6 shows that this approach has led to the best throughput at each channel SNR. Note that the two design metrics seem to be conflicting with each other. Also shown here is the throughput of the original PBRL code [2] when its incremental packets arrive in order. Since the ensemble of this code has the best possible threshold at each rate, the throughput of this code when its increments are appended in order is the maximum that is possible for a PBRL code. Our code for metric 2 comes close to the best throughput at many channel SNR values.

IV. Conclusion

We designed rate-compatible codes with universally good performance for any increment ordering. The paper used two

TABLE IV
Average RCA Threshold (E_b/N_0, dB) at Each Rate for Ensemble in (5) Compared Against Original PBRL Ensemble in [2].

Rate	8/11	8/12	8/13	8/14	8/15	8/16
Avg. Thr., (5)	1.717	1.328	0.998	0.727	0.515	0.394
cf. Table I, (2)	1.809	1.49	1.172	0.86	0.568	0.308

design metrics to obtain such codes, which are called UIO codes. One metric requires, at each rate, the same performance for all increment orderings. The other metric sacrifices identical performance for every ordering to seek codes that have the best average threshold at each rate.

References

[1] R. G. Gallager, "Low-Density Parity-Check Codes," 1963.
[2] T.-Y. Chen, K. Vakilinia, D. Divsalar, and R. Wesel, "Protograph-Based Raptor-Like LDPC Codes," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1522–1532, May 2015.
[3] M. R. Yazdani and A. H. Banihashemi, "On Construction of Rate-Compatible Low-Density Parity-Check Codes," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 159–161, Mar. 2004.
[4] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, "Rate-Compatible Punctured Low-Density Parity-Check Codes With Short Block Lengths," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 728–738, Feb. 2006.
[5] T. V. Nguyen, A. Nosratinia, and D. Divsalar, "The Design of Rate-Compatible Protograph LDPC Codes," *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2841–2850, Oct. 2012.
[6] H. Zeineddine and M. M. Mansour, "Inter-Frame Coding For Broadcast Communication," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 437–452, Feb. 2016.
[7] J. Thorpe, "Low-Density Parity-Check (LDPC) Codes Constructed from Protographs," JPL, IPN-PR 42-154, Aug. 2003.
[8] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
[9] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, "Capacity-Approaching Protograph Codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.
[10] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
[11] T. Tian, C. Jones, J. Villasenor, and R. Wesel, "Construction of Irregular LDPC Codes with Low Error Floors," in *Proc. IEEE Int. Conf. Commun.*, vol. 5, May 2003, pp. 3125–3129.